# Reinforcement Learning is So Confusing

Tri Nguyen
nguyetr9@oregonstate.edu

June 7, 2024

I've read this paper (Madjiheurem et al. 2021) a long time ago. The authors motivate their new type of updates with this line of reasoning. The well-known TD-update only update value of the immediately preceding states (given if the reward is not zero). The eligibility trace improves upon this by making the update propagating back multiple states along the observed trajectory. Then they proposed that we can even make a step further by not only updating all states along observed trajectory but also all "plausible" counterfactual trajectories.

Although the reasoning is nicely motivated, I am not so convinced and want to know why or how do we have all these kinds of updating. There should be a principle that they are all based on which the author seemingly assume all audience are aware of. I think the origin problem starts with the TD-update. Let's trace back from it. Actually, this note starts from the very beginning of introducing Reinforcement Learning (RL).

**RL Goal.** While the ultimate goal of using RL might be abstract and unqualitative, one must explicitly provide some objective so that we can have a certain of how good/bad are we doing. One of the such common goal is discounted cumulative reward. Be aware of that there are others goal, the reason of choosing this is out of scope for now. It is defined as

$$G_\pi = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$$

where the expectation is taken over all random variables. So we want to maximize a sum of all rewards, including the immediate reward with higher weight and future rewards with less significant weights, and taking an average over multiple runs.

$$\underset{\pi}{\text{minimize}} \quad G_\pi$$

**Block 1.** The goal does not concern with variance, which means it might output a good policy in terms of average but might be very very unstable, e.g., getting a very high $G$ on a particular run, and moderate $G$ for other 99 runs.

Immediately, one could ask: What are challenges of the optimization problem above? It looks like an unconstrained problem, how hard it could be? At the moment, it is infeasible due to the expectation. But hey, all supervised learning problems involve expectation and we've cracked them like eating noodle. In statistical learning, under supervised classification setting, it is guaranteed that if we do well on empirical loss, then the true loss (involving expectation) would be okay. That said, although ultimate goal is true loss, we have a surrogate function containing no expectation to work on. Hmm, then what would be a surrogate function for $G_\pi$? Hold this thought, I'd like to come back later.

**Notation.** We use $S_i$ as a random variable of state at time step $t$, $s_i$ as a particular state in a set of states $\mathcal{S}, |\mathcal{S}| = N$. It is really important to note that $S_i, i \to \infty$ while $s_i, 1 \leq i \leq N$. Similar, $A_i$ is used as a random variable of action at time step $i$, $a_i$ as a particular action in a set of actions $\mathcal{A}, |\mathcal{A}| = M$; $r(S_i, A_i) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ as a reward received at time step $i$;

A way to decompose this $G$ is

$$\begin{aligned} G_\pi &= \mathbb{E}\left[r(S_0, A_0) + \gamma r(S_1, A_1) + \gamma^2 r(S_2, A_2) + \ldots + \gamma^t r(S_t, A_t) + \ldots\right] \\ &= \mathbb{E}\left[r(S_0, A_0) + \gamma \underbrace{\mathbb{E}\left[r(S_1, A_1) + \ldots + \gamma^{t-1} r(S_t, A_t) + \ldots\right]}_{\text{this term is still a RV}}\right] \\ &= \mathbb{E}[r(S_0, A_0) + \gamma \underbrace{V(1, S_1)}_{\text{a RV}}], \end{aligned}$$

where we define something called *value function* as follow.

**Definition 1** (Value function)**.** A value function of a policy is a real-value function, $V_\pi : \mathcal{S} \to \mathbb{R}$, and is defined as

$$V_\pi(i, s) := \mathbb{E}\left[r(s, A_i) + \gamma r(S_{i+1}, A_{i+1}) + \ldots + \gamma^{t-1} r(S_t, A_t) + \ldots | S_i = s\right] \tag{1}$$

One immediate question is whether this quantity is finite, i.e., $V_\pi(i, s) < \infty, \forall i, s$? Roughly speaking, it should be bounded, since $r(s, a) < \infty, \forall s, a$, hence it is upper bounded by harmonic series, which is finite. And of course, the expectation being finite is followed. The conditioning $S_i = s$ is necessary since all remaining RVs $A_i, S_{i+1}, A_{i+1}, \ldots$ depend on that information (although do not show the dependence explicitly)[1].

Then, next question is if the index $i$ is a necessary parameter of $V_\pi$. It *seems* not: considering infinite horizon, index $i$ doesn't matter, i.e., let $i' = i + k, k > 0$,

$$V_\pi(i', s) = V_\pi(i', s) \tag{2}$$

We will show evidence of this later, but short answer is this holds (in the limit).

*Remark* 1. Value function at a state $s$ at particular timestep $i$, i.e., $V_\pi(i, s)$ is a deterministic quantity.

*Remark* 2. Value function is time invariant, i.e., $V_\pi(i, s) = V_\pi(i', s)$ for any $i, i' \geq 0$, considering that there are infinitely many more timesteps after the timestep $\max(i, i')$. Since it is time invariant, we will use $V_\pi(s)$ as convention.

*Remark* 3. Value function of a state can be described by value function of other states.

$$\begin{aligned} V_\pi(1, s) &= \mathbb{E}[r(s, A_1) + \gamma r(S_2, A_2) + \ldots + \gamma^{t-1} r(S_t, A_t) + \ldots | S_1 = s] \\ &= \mathop{\mathbb{E}}_{A_1}\left[r(s, A_1) + \gamma \mathbb{E}[r(S_2, A_2) + \ldots + \gamma^{t-1} r(S_t, A_t) + \ldots] \mid S_1 = s\right] \quad \text{(splitting RVs)} \\ &= \sum_{i=1} \Pr(A_1 = a_i | S_1 = s)\left(r(s, a_i) + \gamma \mathbb{E}[r(S_2, A_2) + \ldots + \gamma^{t-1} r(S_t, A_t) + \ldots \mid S_1 = s, A_1 = a_i]\right) \end{aligned}$$

[1] To demonstrate, you will see the following quantity makes no sense

$$\mathbb{E}\left[r(s, A_1) + \gamma r(S_2, A_2) + \ldots + \gamma^{t-1} r(S_t, A_t) + \ldots\right]$$

The last term is

$$\begin{aligned}
&\mathbb{E}[r(S_2, A_2) + \gamma r(S_3, A_3) + \ldots | S_1 = s, A_1 = a_i] \\
&= \sum_{j=1} \Pr(S_2 = s_j | S_1 = s, A_1 = a_i) \mathbb{E}\left[r(s_j, A_2) + \gamma r(S_3, A_3) + \ldots | S_1 = s, A_1 = a_i, S_2 = s_j\right] \\
&= \sum_{j=1} \Pr(S_2 = s_j | S_1 = s, A_1 = a_i) \mathbb{E}\left[r(s_j, A_2) + \gamma r(S_3, A_3) + \ldots | S_2 = s_j\right] \quad \text{(Markov property)} \\
&= \sum_{j=1} \Pr(S_2 = s_j | S_1 = s, A_1 = a_i) V_\pi(2, s_j)
\end{aligned}$$

Combine those,

$$V_\pi(1, s) = \sum_{i=1} \Pr(A_1 = a_i | S_1 = s) \left( r(s, a_i) + \gamma \sum_{j=1} \Pr(S_2 = s_j \mid S_1 = s, A_1 = a_i) V_\pi(2, s_j) \right)$$

With deterministic policy, the outer sum reduces to a single quantity, (well, when shall we do stochastic?)

$$V_\pi(1, s) = r(s, \pi(s)) + \gamma \sum_{j=1} \Pr(S_2 = s_j \mid S_1 = s, A_1 = \pi(s)) V_\pi(2, s_j)$$

**Block 2.** For deterministic policy,

$$V_\pi(1, s) = r(s, \pi(s)) + \gamma \sum_{j=1} \Pr(S_2 = s_j \mid S_1 = s, A_1 = \pi(s)) V_\pi(2, s_j)$$

We can describe this in a more compactly using matrix/vector notation. Define the following quantities

$$\begin{aligned}
\boldsymbol{v}_1 &= [V_\pi(1, s_1), V_\pi(1, s_2), \ldots, V_\pi(1, s_N)]^\top \in \mathbb{R}^N, \\
\boldsymbol{v}_2 &= [V_\pi(2, s_1), V_\pi(2, s_2), \ldots, V_\pi(2, s_N)]^\top \in \mathbb{R}^N, \\
\boldsymbol{P}_{12} &= \begin{bmatrix} \Pr(S_2 = s_1 | S_1 = s_1, A_1 = \pi(s_1)) & \ldots & \Pr(S_2 = s_N \mid S_1 = s_1, A_1 = \pi(s_1)) \\ \vdots & \vdots & \vdots \\ \Pr(S_2 = s_1 | S_1 = s_N, A_1 = \pi(s_N)) & \ldots & \Pr(S_2 = s_N | S_1 = s_N, A_1 = \pi(s_N)) \end{bmatrix} \in \mathbb{R}^{NM \times N}, \\
\boldsymbol{r}_1 &= [r(s_1, \pi(s_1)), r(s_2, \pi(s_2)), \ldots, r(s_N, \pi(s_N))]^\top \in \mathbb{R}^N
\end{aligned}$$

Notice that all subscripts above is used for timestep:

- $\boldsymbol{v}_1, \boldsymbol{v}_2$ are values at timestep 1 and 2;
- $\boldsymbol{P}_{12}$ are transition matrix from timestep 1 to timestep 2. $\boldsymbol{P}_{12}$ depends on environment's property and the policy. That makes sense since value function depends on the policy $\pi$. However, since both transition defined by environment and the policy is time invariant, $\boldsymbol{P}_{12}$ is the same as $\boldsymbol{P}_{i(i+1)}$ for any $i$. For that reason, let just use $\boldsymbol{P}$.
- $\boldsymbol{r}_1$ is reward at timestep 1. Similarly, it is timestep invariant, let's use $\boldsymbol{r}$.

Then, Block 2 can be written as

$$\boldsymbol{v}_1 = \boldsymbol{r} + \gamma \boldsymbol{P} \boldsymbol{v}_2,$$

and it holds for any timestep $i$,

$$\boldsymbol{v}_i = \boldsymbol{r} + \gamma \boldsymbol{P} \boldsymbol{v}_{i+1},$$

(Well, it takes forever to reach the contraction operator :( , but here we come).

Define a linear[1] operator $T_\gamma : \mathbb{R}^N \to \mathbb{R}^N$ as

$$T_\gamma(\boldsymbol{v}) \triangleq \boldsymbol{r} + \gamma \boldsymbol{P} \boldsymbol{v}$$

Hence,

$$\boldsymbol{v}_i = T_\gamma(\boldsymbol{v}_{i+1})$$

Why does it look unnatural (opposite direction is more natural)? Funny enough, this suggests that we should start from the tail and then go backward to $\boldsymbol{v}_1$. And it is actually what people do.

Okay, in terms of theory, it is okay to go backward, just need to assume index can be negative and goes to negative infinite. The important thing is $T_\gamma$ is a $\gamma$-contractor. That means, going backward far enough, i.e., $T \ll 0$,

$$\|\boldsymbol{v}_T - \boldsymbol{v}_{T+1}\|_\infty \approx 0$$

This also confirms the speculation in (2).

Okay, we figured out some thing about the values function. But still this shreds no light into how to optimize $G_\pi$. In fact, it is now more confusing of how all these thing relate to $G_\pi$.

Hmm, we need another start: A starting point from control theory. Every luckily, we are pointed to a very good direction: (Bertsekas 2012) !!!

**Some other very vague and unorganized thoughts.**

- Consider deterministic policy, all the randomness in $G_\pi$ are from environment (transition matrix mostly). So if we can estimate these randomness, we could solve this optimization as a linear programming, couldn't we?

- The RL problem has 2 interacting parts: estimating the dynamics of environment, find the best policy. One can solve each sub-problem independently, or in a more involving way. I believe that nature of dealing with 2 sub-problem simultaneously is what distinguishes RL from other learning problems. Let's call these problems *Estimation* and *Control*, resp.

- In the beginning of chapter 5.2, one of the discussing issues is that some $(s, a)$ is never visited, hence values at these are hard to estimate, hence one needs exploration so that every possible place is visited. But I'm wondering, is it a generalization issue as in supervised learning?

- The problem formulation itself is not a typical optimization problem: the objective function evolves over times. So it is a completely different problem class. Compared to classical optimization problem, we are solving many different problems, each at one timestep[2], which is parameterized by the state and the optimization variable is the action. Did people try to learn the reward function using neural network as in supervised learning setting?

[1] is it linear?

[2] of course, they should relate to each other in someway

# References

Bertsekas, Dimitri (2012). *Dynamic programming and optimal control: Volume I*. Vol. 1.

Madjiheurem, S et al. (2021). “Expected eligibility traces”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. Association for the Advancement of Artificial Intelligence (AAAI).